Indonesian Journal of Electrical Engineering and Computer Science
Vol. 27, No. 1, July 2022, pp. 431~441
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v27.i1.pp431-441

Review on the parameter settings in harmony search algorithm applied to combinatorial optimization problems

Bilal Ahmed[1], Hazlina Hamdan[1], Abdullah Muhammed[2], Nor Azura Husin[1]
[1]Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Seri Kembangan, Malaysia
[2]Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Seri Kembangan, Malaysia

Article Info

Article history:

Received Jan 7, 2022
Revised May 19, 2022
Accepted May 26, 2022

Keywords:

Combinatorial optimization
Harmony search algorithm
Metaheuristic
Modern nature inspired algorithm
Population-based metaheuristic

ABSTRACT

Harmony search algorithm (HSA) is relatively considered as one of the most recent metaheuristic algorithms. HSA is a modern-nature algorithm that simulates the musicians' natural process of musical improvisation to enhance their instrument's note to find a state of pleasant (harmony) according to aesthetic standards. Lots of variants of HSA have been suggested to tackle combinatorial optimization problems. They range from hybridizing some components of other metaheuristic approaches (to improve the HSA) to taking some concepts of HSA and utilizing them to improve other metaheuristic methods. This study reviews research pertaining to parameter settings of HSA and its applications to efficiently solve hard combinatorial optimization problems.

Corresponding Author:

Bilal Ahmed
Department of Computer Science, Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia
43400 UPM Serdang, Malaysia
Email: al-mashhadani.bilal@student.upm.edu.my

1. INTRODUCTION

Combinatorial optimization problem (COP) is the most important class of optimization problems, which is involved in finding the best solution amongst discrete group of all available ones for a specified problem instance [1], [2]. During the past years, numerous optimization algorithms have been suggested to tackle diverse combinatorial optimization problems efficiently [1]. One of the relatively recent population-based metaheuristic (PBM) algorithms is harmony search algorithm (HSA), which managed to solve several hard combinatorial optimization problems. HSA, which was proposed by Geem *et al*. [3] is a modern nature inspired that has gained increasing attention from many researchers as a state-of-the-art algorithm designing a solution for hard COPs [4]. HSA is inspired by a resemblance to the musical improvisation process taken by musicians to improvise their musical tones to find, through repeated iterations, a state of pleasant/harmony (optimum solution to a problem), according to the aesthetic norm, just like the optimization process attempts to get an optimal solution, which is determined by an objective function. During the music production progression, a musician picks and accumulates various amount of pitches from the preceding knowledge, and performs notes by musical instruments in order to get an ideal state of harmony in music [4]. HSA has its own unique features, which makes it distinct from other metaheuristic algorithms. These features can be summarized as shown in: i) HSA is considered as being simple and flexible, thus it stimulates researchers and academics to perform several modifications to improve its performance. ii) HSA is considered as a

population-based algorithm, but it possesses components of a single-based algorithm. iii) HSA is capable of constructing a new solution (new vector) out of a combination of each current solutions (vectors) (namely, the harmonies inside the HM). This makes the HSA utilize the harmony memory more efficiently than the genetic algorithm (GA), which combines only two parents of solution vectors to form the new solutions [5]. This feature makes HSA free from the building block theory that highly affects the mechanism of GA. iv) HSA is able to consider all component variables in a vector independently; whereas the GA is incapable of doing it, as it has to keep the gene's structure [6]. v) HSA includes a lot of characteristics of existing metaheuristic algorithms [7]. It is capable of preserving a record of former vectors (through HM) in a way resembling the tabu search (TS) algorithm [8]. Likewise, it is capable of varying the parameters of the optimization dynamically, like simulated annealing [9]. Finally, HSA can manage numerous solution vectors concurrently in a way analogous to the genetic algorithm [10]. vi) HSA needs less computational effort, in terms of memory and runtime (unlike GA), as it does not require crossover and mutation operators [5]. vii) More importantly, HSA provides a possibility to attain a balance between exploitation and exploration via tuning the HSA improvisation operators (memory consideration, pitch adjustment, and random selection) [6].

Ever since the advent of the HSA basics in 2001, numerous variants of this algorithm have commenced to solve various kinds of optimization problems. These variants range from hybridizing some components of other metaheuristic methods to the framework of HSA, to taking some concepts of HSA and hybridizing them in the framework of other metaheuristic methods. HSA has been effectively utilized to solve lots of optimization problems. Lots of researches have been conducted in different areas such as timetabling, scheduling, artificial neural network, image processing, vehicles routing, medical imaging, job shop scheduling, and many others. Good references for the applications and developments of HSA are available in [5], [6], [11]. The enhancement of the fundamental HSA has interested lots of scholars to develop and improve the performance of the HSA to produce good quality results that meet the requirements of the problems to be solved. Like other metaheuristic algorithms, basic HSA requires proper parameter setting so as to improve the search performance. Slow convergence problems can arise owing to an improper setting of the parameters. Depending on the problem at hand, these settings need to be carefully assigned. Hence, the empirical experiments trials seem to be the only way to select the best parameter values. In this paper, scholars review the fundamental HSA and the most important hybridizations and modifications found in the literature of HSA.

2. CANONICAL HARMONY SEARCH ALGORITHM

Like a group of musicians, when they develop their harmonies practice by practice, HSA enhances solutions iteration by iteration using a good candidate of solutions that been discovered during the construction of solutions and improvises (generates) a new solution (harmony) iteratively for a given problem. In each iteration, HSA improvises a new solution using the three rules known as i) memory consideration, ii) pitch adjustment, and iii) random consideration [12]. The resemblance between the improvisation of music and optimization problems is illustrated in Table 1.

Table 1. The similarity between optimization and improvisation [13]

Musical terms	Optimization terms
Aesthetic standard	Objective function
Harmony	Solution vector
Improvisation	Generation
Musician	Decision variable
Pitch	Value
Pitch range	Value range
Pleasing harmony	Optimal solution
Practice	Iteration

As illustrated in Table 1, there is some equivalence between the musical terms and optimization. The improvisation in music matches generation of solution (i.e., harmony); every musician or instrument matches every decision variable; every tone (pitch) of music instrument matches the value of the decision variable; the pitch range resembles the range of each variable; the esthetic standard of a harmony matches the value of the objective function of a solution; the repetitively improving of pleasant musical harmony matches improving a solution vector; a new harmony which is generated thru using each of the music instruments matches the optimization problem's solution.

2.1. Fundamental structure of HSA

As pointed out by Moon *et al*. [14], Zhang and Geem [15], and Doush *et al*. [16], the basic procedure of HSA comprises six steps that are applied in order to attain improving process. Step 1: initialize parameters of HSA; step 2: initialize HM; step 3: improvise new harmony; step 4: update HM; step 5: check stopping criteria; and step 6: cadenza. The flowchart shown in Figure 1 explains the basic HSA steps.

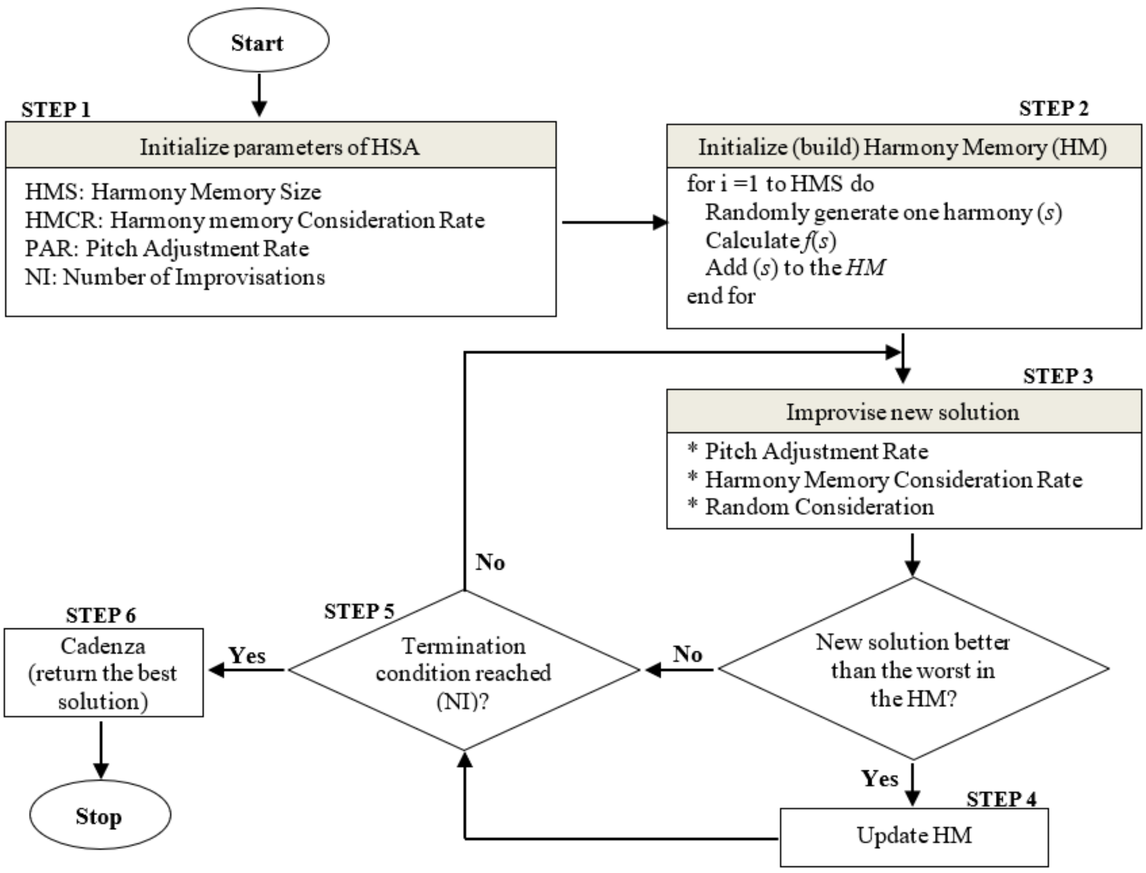

Figure 1. Flowchart of the basic HSA's components

2.1.1. Step 1: Initialize HSA parameters

In this step, the parameters' values of the HSA are initialized. HSA has four parameters [6]:

- Harmony memory size (HMS) defines the total solutions (harmonies) which are kept in the (HM).
- Harmony memory consideration rate (HMCR) is exploited in the improvisation (generation of solution) process. Based on HMCR, the decision variable of a new solution will be chosen from the HM (with HMCR probability) or should be chosen randomly from the available search space (with 1- HMCR probability). HMCR typically takes a range value set between (0, 1).
- Pitch adjustment rate (PAR) is exploited as well in the improvisation process. If a number generated at random is smaller than the value of PAR, the value of the decision variable that has been chosen from the harmony memory should be modified to its neighboring value, otherwise, it has not changed. The PAR value is set between (0, 1).
- Number of improvisation (NI) denotes the maximum value of iterations, which HSA will be repeated, and it is considered as a stopping condition.

2.1.2. Step 2: initialize (build) the harmony memory (HM)

Harmony memory is represented by a 2-D array. It consists of randomly generated solutions, wherein every row denotes a solution vector. The number of generated solutions is equal to HMS and sorted based on the objective function (maximizing or minimizing), as shown in (1).

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_1^2 & \dots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \dots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \dots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix} \begin{bmatrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^{HMS-1}) \\ f(x^{HMS}) \end{bmatrix} \quad (1)$$

where x_1, x_2,…, x_{N-1}, x_N represent a candidate solutions, N represents the number of candidate solutions in each solution vector, and $f(x^1)$, $f(x^2)$,…, $f(x^{HMS})$ represent the corresponding penalty, objective function, of each solution vector.

2.1.3. Step 3: Improvise a new harmony (solution)

During this stage, a new harmony (solution) vector, $x' = (x'1, x'2, x'3, \dots, x'N)$, is generated (improvised) via the improvisation process according to the three rules known as: i) memory consideration, ii) pitch adjustment, and iii) random consideration [6]. As shown in (2) shows the selection mechanism of HSA.

$$x_i^{New} \leftarrow \begin{cases} x_{i(k)} \in \{x_i(2), \dots, x_i(k_i)\} & \text{w.p } P_{\text{Random}} \\ x_i(k) \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} & \text{w.p } P_{\text{Memory}} \\ x_i(k \pm m) & \text{w.p } P_{\text{Pitch}} \end{cases} \quad (2)$$

As shown in (2), the value of variable i ($i = 1, 2, \dots, j$) would be arbitrarily chosen out of a group of all candidate values $\{xi(1), xi(2), \dots, xi(ki)\}$ with a probability of P_{Random} (random consideration); or it would be chosen out of a group of values kept in the HM with a probability of P_{Memory} (memory consideration); or with a little change by moving to neighbor values $xi(k \pm m)$, with a probability of P_{Pitch} (pitch adjustment) [13].

2.1.4. Step 4: Updating harmony memory (HM)

The improvised harmonies (new solutions) are evaluated according to the objective function $f(x)$ value. If the new harmony is better in terms of quality, the HM is updated by substituting the worse harmony with the improvised one $x' = (x'1, x'2, x'3, .., x'N)$. Then, the HM is sorted so that to allocate the new harmony in the right position. Otherwise, the newly generated solution is ignored.

2.1.5. Step 5: Check stopping criteria

This step examines the HSA termination condition. If the stopping condition happens (reaches the maximum number of improvisations (NI)), the algorithm will stop, or else the updating and improvisation processes in steps three and four are repeated. Generaly, the maximum number of improvisations is treated as the stoping criterion in HSA.

2.1.6. Step 6: Cadenza

A cadenza is a bravura musical performance, which takes place at the last stage of a movement of a composition in order to yield the best pleasant harmony ever performed in the improvisation process. In the sense of the HSA, a cadenza may be denoted as the final stage that happens in the last part of searching procedure for the finest harmony. During this step, the HSA returns the highest-quality solution ever stored and found in the HM according to the fitness function *f(x).*

3. STATE-OF-THE-ART HARMONY SEARCH ALGORITHM

Over the last years, HSA has drawn increasing attention as scholars have endeavored to utilize it to solve various optimization problems. Accordingly, lots of scholars have made an effort to develop and improve the performance of the HSA to get good quality results that meet the requirements of the problems to be solved. In order to improve the search efficiency of basic HSA, it requires appropriate parameter setting; for example, a slow convergence issue might happen because of improper parameter setting. Those settings need to be carefully assigned depending on the problem at hand. Therefore, several scholars have attempted to tackle the slow convergence problem of HSA via conducting several enhancing modifications. Modifications comprised i) Carefully tuning the HSA parameters, and ii) Hybridizing HSA with various metaheuristics [7]. The following subsections highlight the most important modifications that have been made to the HSA over the last two decades.

3.1. Modifications to harmony memory (HM) Initialization

After initializing the parameters of the HSA, the second step is the initialization of the HM, where a set of harmonies (solutions) is created randomly to fill up the HM. The harmony memory size (HMS), which denotes the number of harmonies in the HM, is set in the initialization step. According to the literature of HSA, a few studies have attempted to improve the initialization of the HM.

In order to improve the frequency and get global optima, Geem [8] presents two modifications for varying (diversification) the solutions kept in HM. The first one generates a number of initial solutions that in total are larger than the HM size, and the second restricts the number of similar solutions kept in HM. Hadwan *et al*. [9] utilized a semi-cyclic shift pattern method to initialize the HM, in place of using the random mechanism in classical HSA. Pichpibul and Kawtummachai [10] incorporated the probabilistic savings heuristic into HSA to enhance the HM initialization. In order to achieve selection that is more efficient during the improvisation process, they employ the roulette wheel selection mechanism. Al-Betar *et al*. [17] suggested the island-based HSA (iHS) that has been adopted in evolutionary algorithms (EAs). In iHS, the HM is divided into several subpopulations (islands) and every island evolves individually for a number of repetitions.

3.2. Modifications to improvisation step

During the last few years, many scholars have attracted modifications of the improvisation process of the HSA. The selection of the most suitable parameter setting values is too difficult duty for the HSA, and for many other metaheuristic algorithms as well. The setting of these parameter values is problem-dependent. Hence, empirical experiments are the only way for choosing the best parameter values [18].

Many variants of the HSA have been proposed based on several additional mechanisms, concepts, or components to make the HSA parameters adapt dynamically during the search process. In this context, Wang *et al.* [19] propose an algorithm called the adaptive binary harmony search (ABHS) algorithm to tackle binary-coded optimization problems. In ABHS there are some modifications such as a new rule for PAR that is used to improve the search capability. Also, they applied two new HCMR techniques (the bit selection strategy and the individual selection strategy). In addition, they investigated two updating methods (parallel and serial updating), and they developed a scalable adaptive strategy to enhance its optimization capability and robustness regarding the parameter study. Lin and Li [20] proposed a hybrid binary HSA (HBHSA) to tackle winner determination problem. Both PAR and HMCR are changed to attain the balance between diversification and intensification. Keshtegar *et al*. [21] introduced another variant of HSA called adaptive dynamic harmony search (ADHS) algorithm. In this algorithm, a new solution was improvised according to the information kept in HM. The new HM is generated dynamically in two phases. In the first phase, the HM is adjusted based on the minimum and maximum number of the decision variables with a dynamic HMCR as shown in (3).

$$HMCR\ (i) = 0.95 + 0.1 \times \sqrt{\frac{i}{NI} - \left(\frac{i}{NI}\right)^2} \tag{3}$$

Where, *i* denotes the number of the current iteration, and *NI* denotes the max value of iterations. Based on (4), the harmony memory is computed as:

$$x_i^{\prime j} = \begin{Bmatrix} x_i^{\prime j} \pm \sqrt{1-k/NI} \times BW_i(k), with\ probability\ HMCR\ (k) \\ x_i^L + r \times (x_i^U - x_i^L), with\ probability\ 1 - HMCR\ (k) \end{Bmatrix} \tag{4}$$

where, *BW(k)* is a dynamical bandwidth, that proposed as in (5),

$$BW_i(k) = \frac{x_i^{max} - x_i^{min}}{10} \times e^{(-10\frac{k}{NI})} \tag{5}$$

where x_i^{min} and x_i^{max} are the minimum and maximum values for design variable x_i in HM. In the second step, the new HM was modified by the coefficient and the PAR as described in (6).

$$PAR(k) = 0.3 + 0.6 \times (1 - \sqrt{1 - \frac{k}{NI}}) \tag{6}$$

Mansor *et al*. [22] studied the problem of premature or slow convergence by dynamically changing parameters in HSA. In this work, HMCR is given a small value in the first step of generation. To improve the convergence, Gaussian distribution function is utilized to produce numbers randomly as an alternative to

uniform distribution function. Moreover, the PAR and BW values were changed dynamically. Kalivarapu *et al.* [23] suggested a self-adaptive improved HSA (SIHSA) on the basis of dynamic change in BW. The adjustment in BW is calculated as in (7):

$$BW(gn) = \frac{z}{1+k\left(\frac{gn}{NI}\right)^{m}} \tag{7}$$

$$k = 50 * In\left(\frac{BW_{max}}{100 * BW_{min}}\right) \tag{8}$$

where z and m are constants, and their values count on the minimum and maximum values of BW, and m should be more than 1.

Another variant of HSA for solving continuous optimization problems was suggested by Abedinpourshotorban *et al.* [24]. Two improvement techniques were added to the standard HSA in this variant. The first was applied in the initialization of the harmony memory. Instead of a random generation of the initial harmony memory, the solutions in the harmony memory were initialized vertically in such a way that prevents the algorithm from stacking in the local minimum. The second improvement technique replaced the PAR in the basic HSA with the mutation strategy taken from the differential evolution algorithm. The suggested algorithm is compared with other variants of HSA. The result indicated that the proposed algorithm outperformed the algorithms in terms of convergence rate and robustness.

Wang *et al.* [25] proposed a new HSA variation on the basis of dual strategies and adaptive parameters (DSAHS). In DSAHS, the initial value of the HMCR is fixed to 0.95 and the PAR is set to 0.5. During the search, HMCR and PAR are tuned automatically on the basis of the feedback of the fitness of new harmony, if the current HM is updated, the values are kept unchanged, otherwise, HMCR randomly picks a value in a range of (0:9; 1:0) and PAR randomly selects a value in a range of (0:1; 1:0). Guo *et al*. [26] proposed an adaptive HS with best-based strategy (ABHS). ABHS adopts a similar parameter control strategy with DSAHS [25]. In ABHS, the initial value of the parameter HMCR is generated at random in a range of (0:9; 1:0) and the parameter PAR is generated at random in a range of (0:3; 1:0). During each iteration, HMCR and PAR are re-initialized with a low probability of 0:1.

Shaqfa and Orban [27] changed the PAR step of parameter-setting-free HSA (PSFHSA) proposed by Geem and Sim [28] to improve the problem of the parameter-setting of HSA, and used it on the concrete beam design optimization. To enhance the capability of PSFHSA, Jeong *et al*. [29] improved an enhanced form of PSFHSA (APSFHSA) by reducing the extra memory needed throughout the search progression.

3.3. Modifications to stopping criteria

In the HSA, the improvisation process of a new harmony and the updating of the HM are terminated once the stopping condition is met. Many stopping criteria have been used for the HSA such as: reaching maximum computational time, reaching the maximum number of improvisations, reaching predefined value of the objective function, reaching maximum or minimum results, or reaching a predefined number of non-improvements in the objective function during the improvisation process. A little research can be found in the literature pertaining to proposing a new stopping criterion of the HSA. Kattan *et al*. [30] used the improved harmony search (IHS) (which is suggested by Mahdavi *et al*. [31]), for feed-forward artificial neural networks (ANN) as a new method for training techniques. The authors modified the stopping criterion of the IHS according to best-to-worst (BtW) harmony ratio in the current HM instead of using the standard stopping criterion where HSA halts once the maximum number of improvisations (NI) is reached. Fourie *et al*. [32] proposed another stopping criterion, where they measured the distance between the best and worst solution in the HM. Moreover, they also used a trade-off between speed and accuracy to stop the current process and move to the next phase. Table 2 summarizes the most important works related to the HSA parameters setting.

3.4. Hybridizing the HSA with other metaheuristic components

The process of hybridizing some components of variants of metaheuristic algorithms to enhance the performance of these algorithms has been reported in the literature and is highly suggested by many scholars [5], [18]. HSA is characterized as being simple and flexible, so it motivates scholars to combine it with some other metaheuristic components. Moh'd Alia and Mandava [7] categorized the studies related to hybridization of HSA with other metaheuristics into two groups. The first group focuses on the hybridization of metaheuristic components with HSA to enhance its performance. Whilst the second group focuses on the combination of the components of HSA with other metaheuristics. HSA has been hybridized with other metaheuristic algorithms such as ant colony optimization (ACO) [33], genetic algorithm (GA) [34], particle

swarm optimization (PSO) [35], simulated annealing (SA) [36], memetic algorithm (MA) [37] and many others [6]. Table 3 summarizes the studies related to the hybridization of HSA with other metaheuristics.

Table 2. Variants of HSA on the basis of improvements of parameters setting

Algorithm	Parameter(s)	Description	References
HSA-variant	stopping condition	The termination condition is substituted with best-to-worst (BtW) harmony ratio in the present HM	Kattan *et al*. [30]
EHSA	HMCR, PAR, HM initialization	The values of HMCR and PAR are dynamically changed. A semi-cyclic shift pattern approach is utilized to initiate the HM	Hadwan *et al*. [9]
ABHS	PAR	A new pitch adjustment strategy is adopted to improve the BHS performance via selecting the adjacent value from its neighborhood	Wang *et al.* [19]
HSA-variant	HM initialization	Incorporate the probabilistic savings heuristic into HSA to enhance the HM initialization	(Pichpibul and Kawtummachai [10]
iHS	HM	It learns the concepts of island model that has been adopted in EA.	Al-Betar *et al.* [17]
SIHSA	BW	A self-adaptive Improved HSA (SIHSA) on basis of dynamic change in BW.	Kalivarapu *et al*. [23]
HSA-variant	HMCR, PAR, BW	HMCR is given a small value in the first step of generation. the PAR and BW values were changed dynamically.	Mansor *et al*. [22]
HSA-variant	PAR, HM initialization	Instead of a random generation of the initial HM, the solutions in the HM were initialized vertically. The PAR was replaced with the mutation strategy taken from the differential evolution algorithm.	Abedinpourshotorban *et al*. [24]
ADHS	HM	The new HM is generated dynamically in two phases. In the first phase, the HM is adjusted based on the minimum and maximum number of the decision variables with a dynamic HMCR. In the second stage, the new HM was modified by the coefficient and the PAR	Keshtegar *et al*. [21]
DSAHS	HMCR, PAR	During the search, PAR and HMCR are tuned automatically on the basis of the feedback of the fitness of new harmony.	Wang *et al*. [25]
ABHS	HMCR, PAR	The initial value of the HMCR is generated randomly in a range of (0:9; 1:0) and the parameter PAR is generated randomly in a range of (0:3; 1:0). During each iteration, HMCR and PAR are re-initialized with a low probability of 0:1.	Guo *et al*. [26]
MPSFHSA	PAR	The step of pitch adjustment is modified in PSFHSA	Shaqfa and Orban [27]
HBHSA	PAR, HMCR	Both PAR and HMCR are changed to get the balance between diversification and intensification	Lin and Li [20]
APSFHSA	HMCR, PAR, BW	An enhanced form of PSFHSA (APSFHSA) by reducing the extra memory requirement throughout the improvisation process	Jeong *et al*. [29]

Table 3. Hybridizing HSA with components from other metaheuristic algorithms

Algorithm	Description	References
HSA+BA	The Bee algorithm generates the HM for HSA.	Nguyen *et al*. [38]
ACO+HSA (AntHSA)	The HSA is hybridized with two concepts borrowed from ACO (pheromone and heuristic values).	Amini and Ghaderi [39]
HSA+ RW	The roulette wheel selection mechanism is adopted to achieve more efficient selection during the improvisation process.	Pichpibul and Kawtummachai [10]
HSA+BBO	The mutation operator in the BBO is utilized in the improvisation phase of the HSA.	Wang *et a*l. [40]
HSA+SS+ SU+ MB	The HM is initialized based on reference set mechanism. HSA convergence is improved via including Symmetrical Uncertainty and Markov Blanket filter within HSA.	Shreem *et al*. [41]
HSA+PSO	PSO is used to enhance the solution quality of the initial harmony memory (HM).	Zhao *et al*. [42]
HSA+Global-best PSO	This hybrid algorithm changes the randomization mechanism of HSA with Global-best PSO search and neighborhood search.	Cheng *et al*. [43]
HSA+BBO	To exploit the migration operation of BBO to enhance the new harmony improvisation process of HSA.	Zheng *et al.* [44]
HSA+GDL	The searching ability of GDL is utilized to optimize the weight of FLANN in order to improve the classification accuracy.	Naik *et al.* [45]
HSA+CS	The mutation operator is used to enhance the harmony in HSA via enhancing the convergence speed.	Wang *et al.* [46]
HSA+ACO	The ideas of ACO are combined into HSA to improve the convergence rate.	Fouad *et al.* [33]
HSA+FA	HSA is utilized to mutate the firefly algorithm to escape the solutions, that stuck into local optimum.	Rehman *et al*. [47]
HSA+VNS	A hybrid HSA with variable neighborhood search (VNS) is suggested to tackle permutation flow shop scheduling problems (PFSSP).	Zhao *et al*. [48]
HSA+HC+SA+RTR+RTS	AHSA inserts an adaptive selection mechanism to automatically choose a proper SBM for vehicle routing problem.	Yassen *et al*. [49]
HSA+Jaya	HSA has been adapted to tackle the 0/1 Knapsack problem by combining objective function to treat the weight criterion and excluding the bandwidth of HSA to get better results.	Alomoush *et al*. [50]
HSA+SA	HSA has been modified to accept even inferior solutions with likelihood defined by a parameter known as Temperature (T).	Assad and Deep [51]

On the other hand, metaheuristic algorithms have been given an HSA component to enhance the performance of those algorithms through the integration of the HSA concepts into the structures of these metaheuristics. For instance, Li and Li [52] proposed a Novel hybrid particle swarm optimization (NHPSO) aiming to solving high-dimensional optimization problems in an efficient way. The authors merge the HSA into PSO to avoid the variables from flying outside their variables' boundaries (prevent the *pbest* concept from violating the variables boundary). In another example, the GA performance was developed using the HSA improvisation concept. In this work, the mechanism of selecting the decision variables of the HSA throughout the improvisation process is employed to improve the selection mechanism of the GA [53]. Table 4 summarizes some examples of the hybridization of HSA components with other metaheuristic algorithms.

Table 4: Hybridizing components and concepts of HSA into other metaheuristic algorithms

Algorithm	Description	References
GA+HS	The HSA idea of searching is utilized to enhance the ability of Genetic Algorithm.	Qinghua *et al*. [54]
PSO+HS	The idea of HM in HSA is combined into PSO to avoid the *pbest* concept of PSO from violating the boundary of the variables.	Li *et al*. [52]
GA+HS	The mechanism of choosing the decision variables from all vectors kept in the HM is imitated to enhance the selection mechanism of GA.	Li *et al*. [53]
LDA+HSA	HSA is utilized as a preprocessing method to tackle the Linear Discriminate Analysis (LDA) problem.	Moeinzadeh *et al*. [55]
PSOPC+ ACO+ HS	The concept of HM is utilized to adjust the variable constraints in Particle Swarm Optimizer with Passive Congregation (PSOPC).	Kaveh and Talatahari [56]
GA+HS	HSA is employed to attain the balance between exploitation and exploration in the GA.	Nadi *et al*. [57]
BA+HSA	The PAR is used as a mutation operator throughout the process of the bat algorithm to enhance the convergence speed.	Wang and Guo [58]
CRO+HSA	The HSA improvisation rules are adopted in the reproduction process of the Coral Reefs Optimization algorithm (CRO).	Salcedo-Sanz *et al*. [59]
PSO+HSA	A method for improving network lifetime is suggested via exploiting PSO algorithm-based clustering and HSA-based routing in WSNs.	Anand and Pandey [60]
CS+HSA	A hybrid Cuckoo Search (CS) with HSA-based energy equivalent node clustering protocol is suggested, that uses a new fitness value for the uniform pattern of cluster leaders.	Gupta and Jha [61]

4. CONCLUSION

Harmony search algorithm (HSA) has confirmed its capability in tackling hard combinatorial optimization problems such as scheduling problems, course timetabling, vehicles routing, and many others. HSA still attracts the attention of scholars because of its generality, flexibility, and has a well-balanced mechanism to improve both global and local exploration abilities. However, despite the good performance of HSA, researchers still need to overcome some shortcomings, such as slow convergence due to the HSA's fully random mechanism for generating the initial harmony memory. Different issues and aspects related to HSA have been discussed concerning new variants proposed to overcome the weakness of basic HSA. Many variants of HSA have been proposed by hybridizing some concepts of other metaheuristic approaches to enhance HSA or employing some components of HSA to improve other metaheuristic methods. This study also had reviewed research pertaining to parameter settings in HSA and its applications to efficiently solve hard combinatorial optimization problems.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Universiti Putra Malaysia (UPM) for supporting and funding this work (Putra Grant No.:9569200).

REFERENCES

[1] B. Korte and J. Vygen, *Combinatorial Optimization*, vol. 21. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018.

[2] E. G. Talbi and C. Ribeiro, "Special issue on 'optimization and machine learning,'" *International Transactions in Operational Research*, vol. 25, no. 4, pp. 1407–1407, Jul. 2018, doi: 10.1111/itor.12528.

[3] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, Feb. 2001, doi: 10.1177/003754970107600201.

[4] A. A. Alomoush, A. R. A. Alsewari, K. Z. Zamli, A. Alrosan, W. Alomoush, and K. Alissa, "Enhancing three variants of harmony search algorithm for continuous optimization problems," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2343–2349, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2343-2349.

[5] A. A. Al-Omoush, A. A. Alsewari, H. S. Alamri, and K. Z. Zamli, "Comprehensive review of the development of the harmony search algorithm and its applications," *IEEE Access*, vol. 7, pp. 14233–14245, 2019, doi: 10.1109/ACCESS.2019.2893662.

[6] M. Dubey, V. Kumar, M. Kaur, and T. P. Dao, "A Systematic review on harmony search algorithm: theory, literature, and applications," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–22, Apr. 2021, doi: 10.1155/2021/5594267.

[7] O. M. d. Alia, R. Mandava, and M. E. Aziz, “A hybrid harmony search algorithm for MRI brain segmentation,” *Evolutionary Intelligence*, vol. 4, no. 1, pp. 31–49, Mar. 2011, doi: 10.1007/s12065-011-0048-1.
[8] Z. W. Geem, “Effects of initial memory and identical harmony in global optimization using harmony search algorithm,” *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11337–11343, Jul. 2012, doi: 10.1016/j.amc.2012.04.070.
[9] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, “A harmony search algorithm for nurse rostering problems,” *Information Sciences*, vol. 233, pp. 126–140, Jun. 2013, doi: 10.1016/j.ins.2012.12.025.
[10] T. Pichpibul and R. Kawtummachai, “Modified harmony search algorithm for the capacitated vehicle routing problem,” in *Lecture Notes in Engineering and Computer Science*, 2013, vol. 2203, pp. 1094–1099.
[11] J. Yi, C. Lu, and G. Li, “A literature review on latest developments of harmony search and its applications to intelligent manufacturing,” *Mathematical Biosciences and Engineering*, vol. 16, no. 4, pp. 2086–2117, 2019, doi: 10.3934/mbe.2019102.
[12] J. Gholami, K. K. A. Ghany, and H. M. Zawbaa, “A novel global harmony search algorithm for solving numerical optimizations,” *Soft Computing*, vol. 25, no. 4, pp. 2837–2849, Feb. 2021, doi: 10.1007/s00500-020-05341-5.
[13] Z. Woo *et al*., *Recent Advances in Harmony Search*, vol. 270. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
[14] Y. Y. Moon, Z. W. Geem, and G. T. Han, “Vanishing point detection for self-driving car using harmony search algorithm,” *Swarm and Evolutionary Computation*, vol. 41, pp. 111–119, Aug. 2018, doi: 10.1016/j.swevo.2018.02.007.
[15] T. Zhang and Z. W. Geem, “Review of harmony search with respect to algorithm structure,” *Swarm and Evolutionary Computation*, vol. 48, pp. 31–43, Aug. 2019, doi: 10.1016/j.swevo.2019.03.012.
[16] I. Abu Doush *et al.*, “Flow shop scheduling with blocking using modified harmony search algorithm with neighboring heuristics methods,” *Applied Soft Computing*, vol. 85, p. 105861, Dec. 2019, doi: 10.1016/j.asoc.2019.105861.
[17] M. A. Al-Betar, M. A. Awadallah, A. T. Khader, and Z. A. Abdalkareem, “Island-based harmony search for optimization problems,” *Expert Systems with Applications*, vol. 42, no. 4, pp. 2026–2035, Mar. 2015, doi: 10.1016/j.eswa.2014.10.008.
[18] A. Turky, N. R. Sabar, S. Dunstall, and A. Song, “Hyper-heuristic local search for combinatorial optimisation problems,” *Knowledge-Based Systems*, vol. 205, p. 106264, Oct. 2020, doi: 10.1016/j.knosys.2020.106264.
[19] L. Wang, R. Yang, Y. Xu, Q. Niu, P. M. Pardalos, and M. Fei, “An improved adaptive binary harmony search algorithm,” *Information Sciences*, vol. 232, pp. 58–87, May 2013, doi: 10.1016/j.ins.2012.12.043.
[20] Z. Li and G. Lin, “A hybrid binary harmony search algorithm for solving the winner determination problem,” *International Journal of Innovative Computing and Applications*, vol. 10, no. 1, p. 59, 2019, doi: 10.1504/ijica.2019.10022307.
[21] B. Keshtegar, P. Hao, Y. Wang, and Y. Li, “Optimum design of aircraft panels based on adaptive dynamic harmony search,” *Thin-Walled Structures*, vol. 118, pp. 37–45, Sep. 2017, doi: 10.1016/j.tws.2017.05.004.
[22] N. F. Mansor, Z. A. Abas, A. F. N. A. Rahman, A. S. Shibghatullah, and S. Sidek, “A new HMCR parameter of harmony search for better exploration,” in *Advances in Intelligent Systems and Computing*, vol. 382, 2016, pp. 181–195, doi: 10.1007/978-3-662-47926-1_18.
[23] J. Kalivarapu, S. Jain, and S. Bag, “An improved harmony search algorithm with dynamically varying bandwidth,” *Engineering Optimization*, vol. 48, no. 7, pp. 1091–1108, Jul. 2016, doi: 10.1080/0305215X.2015.1090570.
[24] H. Abedinpourshotorban, S. Hasan, S. M. Shamsuddin, and N. F. As’Sahra, “A differential-based harmony search algorithm for the optimization of continuous problems,” *Expert Systems with Applications*, vol. 62, pp. 317–332, Nov. 2016, doi: 10.1016/j.eswa.2016.05.013.
[25] Y. Wang, Z. Guo, and Y. Wang, “Enhanced harmony search with dual strategies and adaptive parameters,” *Soft Computing*, vol. 21, no. 15, pp. 4431–4445, Aug. 2017, doi: 10.1007/s00500-017-2563-1.
[26] Z. Guo, H. Yang, S. Wang, C. Zhou, and X. Liu, “Adaptive harmony search with best-based search strategy,” *Soft Computing*, vol. 22, no. 4, pp. 1335–1349, Feb. 2018, doi: 10.1007/s00500-016-2424-3.
[27] M. Shaqfa and Z. Orbán, “Modified parameter-setting-free harmony search (PSFHS) algorithm for optimizing the design of reinforced concrete beams,” *Structural and Multidisciplinary Optimization*, vol. 60, no. 3, pp. 999–1019, Sep. 2019, doi: 10.1007/s00158-019-02252-4.
[28] Z. W. Geem and K. B. Sim, “Parameter-setting-free harmony search algorithm,” *Applied Mathematics and Computation*, vol. 217, no. 8, pp. 3881–3889, Dec. 2010, doi: 10.1016/j.amc.2010.09.049.
[29] Y. W. Jeong, S. M. Park, Z. W. Geem, and K. B. Sim, “Advanced parameter-setting-free harmony search algorithm,” *Applied Sciences (Switzerland)*, vol. 10, no. 7, p. 2586, Apr. 2020, doi: 10.3390/app10072586.
[30] A. Kattan, R. Abdullah, and R. A. Salam, “Harmony search based supervised training of artificial neural networks,” in *ISMS 2010 - UKSim/AMSS 1st International Conference on Intelligent Systems, Modelling and Simulation*, Jan. 2010, pp. 105–110, doi: 10.1109/ISMS.2010.31.
[31] M. Mahdavi, M. Fesanghary, and E. Damangir, “An improved harmony search algorithm for solving optimization problems,” *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, May 2007, doi: 10.1016/j.amc.2006.11.033.
[32] J. Fourie, S. Mills, and R. Green, “Harmony filter: A robust visual tracking system using the improved harmony search algorithm,” *Image and Vision Computing*, vol. 28, no. 12, pp. 1702–1716, Dec. 2010, doi: 10.1016/j.imavis.2010.05.006.
[33] A. Fouad, D. Boukhetala, F. Boudjema, K. Zenger, and X. Z. Gao, “A novel global harmony search method based on ant colony optimisation algorithm,” *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 28, no. 1–2, pp. 215–238, Mar. 2016, doi: 10.1080/0952813X.2015.1020570.
[34] S. H. Huang and P. C. Lin, “A harmony-genetic based heuristic approach toward economic dispatching combined heat and power,” *International Journal of Electrical Power and Energy Systems*, vol. 53, no. 1, pp. 482–487, Dec. 2013, doi: 10.1016/j.ijepes.2013.05.027.
[35] M. G. H. Omran and M. Mahdavi, “Global-best harmony search,” *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, May 2008, doi: 10.1016/j.amc.2007.09.004.
[36] J. Sui, L. Yang, H. Fan, and Z. Hua, “Mine airflow optimizing control based on harmony annealing search,” in *Proceedings - 2010 International Conference of Information Science and Management Engineering, ISME 2010*, Aug. 2010, vol. 2, pp. 295–298, doi: 10.1109/ISME.2010.263.
[37] Y. Zhang, P. Lin, Z. Chen, and S. Cheng, “A population classification evolution algorithm for the parameter extraction of solar cell models,” *International Journal of Photoenergy*, vol. 2016, pp. 1–16, 2016, doi: 10.1155/2016/2174573.
[38] K. Nguyen, P. Nguyen, and N. Tran, “A hybrid algorithm of harmony search and bees algorithm for a university course timetabling problem,” *International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 12–17, 2012.
[39] F. Amini and P. Ghaderi, “Hybridization of harmony search and ant colony optimization for optimal locating of structural dampers,” *Applied Soft Computing Journal*, vol. 13, no. 5, pp. 2272–2280, May 2013, doi: 10.1016/j.asoc.2013.02.001.
[40] G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, and M. Shao, “Hybridizing harmony search with biogeography based optimization for global numerical optimization,” *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 10, pp. 2312–2322, Oct. 2013, doi: 10.1166/jctn.2013.3207.

[41] S. S. Shreem, S. Abdullah, and M. Z. A. Nazri, "Hybridising harmony search with a Markov blanket for gene selection problems," *Information Sciences*, vol. 258, pp. 108–121, Feb. 2014, doi: 10.1016/j.ins.2013.10.012.

[42] F. Zhao, Y. Liu, C. Zhang, and J. Wang, "A self-adaptive harmony PSO search algorithm and its performance analysis," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7436–7455, Nov. 2015, doi: 10.1016/j.eswa.2015.05.035.

[43] M. Y. Cheng, D. Prayogo, Y. W. Wu, and M. M. Lukito, "A hybrid harmony search algorithm for discrete sizing optimization of truss structure," *Automation in Construction*, vol. 69, pp. 21–33, Sep. 2016, doi: 10.1016/j.autcon.2016.05.023.

[44] Y. J. Zheng, M. X. Zhang, and B. Zhang, "Biogeographic harmony search for emergency air transportation," *Soft Computing*, vol. 20, no. 3, pp. 967–977, Mar. 2016, doi: 10.1007/s00500-014-1556-6.

[45] B. Naik, J. Nayak, and H. S. Behera, "A global-best harmony search based gradient descent learning FLANN (GbHS-GDL-FLANN) for data classification," *Egyptian Informatics Journal*, vol. 17, no. 1, pp. 57–87, Mar. 2016, doi: 10.1016/j.eij.2015.09.001.

[46] G. G. Wang, A. H. Gandomi, X. Zhao, and H. C. E. Chu, "Hybridizing harmony search algorithm with cuckoo search for global numerical optimization," *Soft Computing*, vol. 20, no. 1, pp. 273–285, Jan. 2016, doi: 10.1007/s00500-014-1502-7.

[47] A. U. Rehman *et al*., "Efficient energy management system using firefly and harmony search algorithm," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 12, 2018, pp. 37–49.

[48] F. Zhao, Y. Liu, Y. Zhang, W. Ma, and C. Zhang, "A hybrid harmony search algorithm with efficient job sequence scheme and variable neighborhood search for the permutation flow shop scheduling problems," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 178–199, Oct. 2017, doi: 10.1016/j.engappai.2017.07.023.

[49] E. T. Yassen, M. Ayob, M. Z. A. Nazri, and N. R. Sabar, "An adaptive hybrid algorithm for vehicle routing problems with time windows," *Computers and Industrial Engineering*, vol. 113, pp. 382–391, Nov. 2017, doi: 10.1016/j.cie.2017.09.034.

[50] A. Alomoush, A. A. Alsewari, H. S. Alamri, and K. Z. Zamli, "Solving 0/1 Knapsack Problem Using Hybrid HS and Jaya Algorithms," *Advanced Science Letters*, vol. 24, no. 10, pp. 7486–7489, Oct. 2018, doi: 10.1166/asl.2018.12964.

[51] A. Assad and K. Deep, "A Hybrid Harmony search and Simulated Annealing algorithm for continuous optimization," *Information Sciences*, vol. 450, pp. 246–266, Jun. 2018, doi: 10.1016/j.ins.2018.03.042.

[52] H. Q. Li and L. Li, "A novel hybrid particle swarm optimization algorithm combined with harmony search for high dimensional optimization problems," in *Proceedings The 2007 International Conference on Intelligent Pervasive Computing, IPC 2007*, Oct. 2007, pp. 94–97, doi: 10.1109/IPC.2007.22.

[53] M. J. Li, M. K. Ng, Y. M. Cheung, and J. Z. Huang, "Agglomerative fuzzy K-Means clustering algorithm with selection of number of clusters," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1519–1534, Nov. 2008, doi: 10.1109/TKDE.2008.88.

[54] Q. Li, S. Yang, and Y. Ruan, "Hybrid algorithm for optimizing multi-modal functions," *Wuhan University Journal of Natural Sciences*, vol. 11, no. 3, pp. 551–554, May 2006, doi: 10.1007/BF02836663.

[55] H. Moeinzadeh, E. Asgarian, M. Zanjani, A. Rezaee, and M. Seidi, "Combination of harmony search and linear discriminate analysis to improve classification," in *Proceedings - 2009 3rd Asia International Conference on Modelling and Simulation, AMS 2009*, 2009, pp. 131–135, doi: 10.1109/AMS.2009.125.

[56] A. Kaveh and S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Computers and Structures*, vol. 87, no. 5–6, pp. 267–283, Mar. 2009, doi: 10.1016/j.compstruc.2009.01.003.

[57] F. Nadi, A. T. Khader, and M. A. Al-Betar, "Adaptive genetic algorithm using harmony search," in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, GECCO '10*, 2010, pp. 819–820, doi: 10.1145/1830483.1830628.

[58] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, pp. 1–21, 2013, doi: 10.1155/2013/696491.

[59] S. Salcedo-Sanz, A. Pastor-Sánchez, J. Del Ser, L. Prieto, and Z. W. Geem, "A coral reefs optimization algorithm with harmony search operators for accurate wind speed prediction," *Renewable Energy*, vol. 75, pp. 93–101, Mar. 2015, doi: 10.1016/j.renene.2014.09.027.

[60] V. Anand and S. Pandey, "Particle swarm optimization and harmony search based clustering and routing in Wireless Sensor Networks," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, p. 1252, 2017, doi: 10.2991/ijcis.10.1.84.

[61] G. P. Gupta and S. Jha, "Integrated clustering and routing protocol for wireless sensor networks using Cuckoo and Harmony Search based metaheuristic techniques," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 101–109, Feb. 2018, doi: 10.1016/j.engappai.2017.11.003.

BIOGRAPHIES OF AUTHORS

Bilal Ahmed received his BSc degree in Computer Engineering from University of Technology, Baghdad, Iraq in 2002, and his MSc degree in Information Technology (IT) from Universiti Tenaga Nasional (UNITEN), Putrajaya, Malaysia in 2013. Currently he is pursuing PhD in Faculty of Computer Science & Information Technology, Universiti Putra Malaysia (UPM). His research interests include computational intelligence, metaheuristic optimization algorithms specifically Harmony Search Algorithm (HSA), hyperheuristics and combinatorial optimization problems. He can be contacted at email: GS58255@student.upm.edu.my.

Dr. Hazlina Hamdan received the Ph.D. degree in Artificial Intelligence from the University of Nottingham, UK, in 2013. She is currently a Senior Lecturer with the Department of Computer Science, Universiti Putra Malaysia. She has led several research projects funded by the ministry of higher education (MOHE), and research university grant scheme (GP), and currently leading an on-going two research projects, while a co-researcher in other three projects. Her research areas include intelligent computing and application, such as medical prognostic, pattern recognition, prediction systems, and optimization. She can be contacted at email: hazlina@upm.edu.my.

Dr. Abdullah Muhammed is an Associate Professor at the Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. He received the Bachelor degree in Computer Science from Universiti Putra Malaysia in 1998, the Master degree in Computer Science from Universiti Malaya in 2004 and the PhD degree in Computer Science from University of Nottingham, United Kingdom, in 2014. His research interests include grid/cloud computing, Wireless Sensor Network/IoT, heuristic and optimisation. For more information, please email him at abdullah@upm.edu.my.

Dr. Nor Azura Husin received her PhD in artificial intelligent from Universiti Putra Malaysia, Malaysia in 2013. She is currently a senior lecturer at the Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. Her areas of interest in research include data mining, machine learning, and intelligent computing. Dr. Nor Azura's expertise is in the development of prediction models, and she has mainly contributed to the fields of medicine and agriculture for dengue outbreak prediction and paddy disease detection. She has worked on a consulting project for the design of a web-based and mobile-based online logistics aggregation business. She has also served as a consultant for the Lembaga Hasil Dalam Negeri Malaysia (LHDNM) on an efficacy study about the implementation of subsistence allowance. She can be contacted at email: n_azura@upm.edu.my.